

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-237179

(43)Date of publication of application : 23.08.1994

(51)Int.Cl.

H03M 7/30
G06F 15/66
G10L 9/18
H04N 1/415
H04N 7/13

(21)Application number : 05-260908

(71)Applicant : NIPPON TELEGR & TELEPH
CORP <NTT>

(22)Date of filing : 19.10.1993

(72)Inventor : UEDA SHUKO

(30)Priority

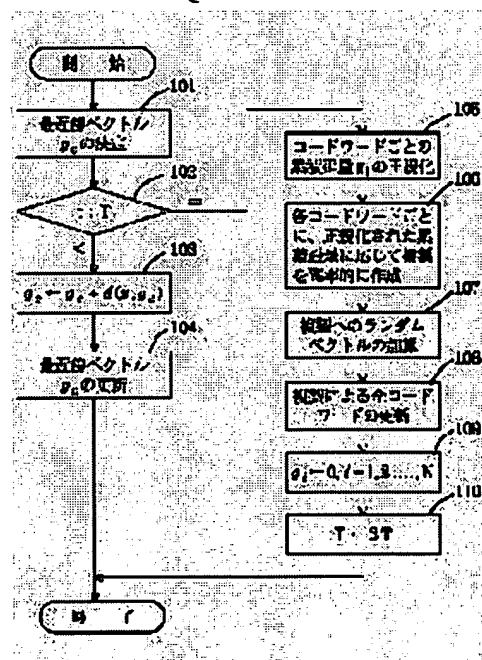
Priority number : 04321976 Priority date : 01.12.1992 Priority country : JP

(54) DESIGN METHOD FOR VECTOR QUANTIZER AND VECTOR QUANTIZER

(57)Abstract:

PURPOSE: To provide the method minimizing average distortion regardless of signal distribution in the design of a code book.

CONSTITUTION: A code word is 'copied' or 'lost' based on the 'principle of equal distortion'. For example, when each code word is selected at a closest vector, a distortion $d(x, y)$ is integrated (103), and when an input data number (t) reaches a number T (101), an adaptive degree is calculated (105) in response to an accumulated distortion $g1(=\sum d(x, y))$, a copy of the relevant code word is generated in terms of probability in response to the adaptive degree (106) and a new code word group is obtained by adding a random vector to each copy respectively (107) and the code book is revised by the code word group (108).



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-237179

(43)公開日 平成6年(1994)8月23日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
H 0 3 M 7/30	B	8522-5 J		
G 0 6 F 15/66	3 3 0 F	8420-5 L		
G 1 0 L 9/18	E	8946-5 H		
H 0 4 N 1/415		9070-5 C		
7/13	Z			

審査請求 未請求 請求項の数7 O L (全 14 頁)

(21)出願番号 特願平5-260908

(22)出願日 平成5年(1993)10月19日

(31)優先権主張番号 特願平4-321976

(32)優先日 平4(1992)12月1日

(33)優先権主張国 日本(J P)

(71)出願人 000004226

日本電信電話株式会社

東京都千代田区内幸町一丁目1番6号

(72)発明者 上田 修功

東京都千代田区内幸町一丁目1番6号 日

本電信電話株式会社内

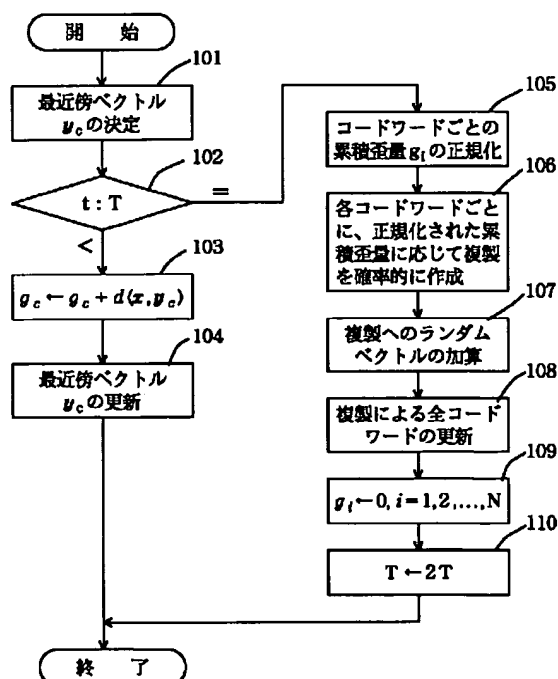
(74)代理人 弁理士 若林 忠

(54)【発明の名称】 ベクトル量子化器の設計方法およびベクトル量子化器

(57)【要約】

【目的】 コードブック設計に際し、信号分布の如何によらず、平均歪みを最小化できる方法を提供する。

【構成】 “等歪み原理”に基づいて、コードワードの“複製”と“消滅”が行なわれるようにする。例えば、各コードワードごとにそのコードワードが最近傍ベクトルとして選択されたときに歪み $d(x, y_c)$ を積算し(103)、入力データ数 t がある数 T に達したとき(101)、各コードワードごとの累積歪量 $g_i (= \sum d(x, y_c))$ に応じて適応度を算出し(105)、適応度に応じて当該コードワードの複製を確率的に作成し(106)、各複製にそれぞれランダムベクトルを加算(107)したものをもって新たなコードワード群として、このコードワード群でコードブックを更新する(108)。



【特許請求の範囲】

【請求項1】 予め有限個のコードワードからなるコードブックが用意され、ある確率分布にしたがう有限個の訓練ベクトルデータの各々に対し前記訓練ベクトルデータとの歪みが最小になるコードワードを前記コードブックの中から選択して最近傍ベクトルとする選択工程と、前記最近傍ベクトルと前記歪みに基づいて前記コードブック内のコードワードを逐次修正する修正工程とを有し、前記確率分布に対応する前記コードブックの設計を行なうベクトル量子化器の設計方法において、前記選択工程では、前記各コードワードの各々に対して、前記最近傍ベクトルとして選択された際の前記歪みの累積値を測定し、前記修正工程に、前記各コードワードごとに当該コードワードについての前記歪みの累積値に応じて適応度を算出し前記適応度に応じて当該コードワードの複製を生成する複製生成工程と、前記複製生成工程で生成した前記各複製にそれぞれ微小なランダムベクトルを加算して新たなコードワードとし、前記各コードワードをもって新たな前記コードブックとする更新工程とが付加され、前記複製生成工程および前記更新工程とが、前記訓練ベクトルデータの提示累積個数が所定数に達したときに実行されることを特徴とするベクトル量子化器の設計方法。

【請求項2】 複製生成工程において各コードワードごとに生成される複製の数が当該コードワードについての歪みの累積値に比例した数に確率的な処理を施したものであり、前記複製の総数が前記複製生成工程を行なう前のコードワードの数に等しいものである請求項1に記載のベクトル量子化器の設計方法。

【請求項3】 予め有限個のコードワードからなるコードブックが用意され、ある確率分布にしたがう有限個の訓練ベクトルデータの各々に対し前記訓練ベクトルデータとの歪みが最小になるコードワードを前記コードブックの中から選択して最近傍ベクトルとする選択工程と、前記最近傍ベクトルと前記歪みに基づいて前記コードブック内のコードワードを逐次修正する修正工程とを有し、前記確率分布に対応する前記コードブックの設計を行なうベクトル量子化器の設計方法において、前記修正工程に、前記各コードワードごとに当該コードワードについての部分歪みに応じて適応度を算出し前記適応度に応じて当該コードワードを複製しあるいは消滅させる複製・消滅工程と、前記複製・消滅工程で生成した各複製にそれぞれ微小なランダムベクトルを加算して新たなコードワードとし、前記各コードワードをもって新たな前記コードブックとする更新工程とが付加され、前記複製・消滅工程および前記更新工程とが、前記訓練ベクトルデータの提示累積個数が所定数に達したときに実行されることを特徴とするベクトル量子化器の設計方法。

【請求項4】 複製・消滅工程において各コードワードごとに生成される複製の数が当該コードワードについての部分歪みの巾乗に比例し、前記複製の総数が前記複製・消滅工程を行なう前のコードワードの数に等しいものである請求項3に記載のベクトル量子化器の設計方法。

【請求項5】 請求項3または4に記載のベクトル量子化器の設計方法を利用したクラス分類型ベクトル量子化器の設計方法であって、

前記ベクトル量子化器の設計方法に、画像を同じサイズの矩形画像に分割して該矩形画像を予め定められた複数の画像クラスに分類する分類工程と、前記分類された矩形画像に含まれる各画素値を列挙して得られるベクトルを前記訓練ベクトルデータとする訓練データ作成工程とが付加され、

予め定められた複数の画像クラスごとに前記コードブックが用意され、

前記選択工程が前記画像クラスごとに実行されることにより前記画像クラスごとに前記コードブックの設計がなされるクラス分類型ベクトル量子化器の設計方法。

【請求項6】 有限個のコードワードからなるコードブックを保持するコードブック記憶手段と、逐次的に入力ベクトルが入力し前記入力ベクトルとの歪みが最小になるコードワードを前記コードブックの中から選択して最近傍ベクトルとする選択手段と、前記最近傍ベクトルに基づいて量子化結果を出力する量子化手段と、前記最近傍ベクトルと前記歪みに基づいて前記コードブック内のコードワードを逐次修正する修正手段とを有するベクトル量子化器において、

前記各コードワードの各々について当該コードワードが前記最近傍ベクトルとして選択された際の前記歪みをそれぞれ積算して記憶する加算手段を有し、

前記修正手段が、前記入力ベクトルの提示累積回数が所定数に達したときには、前記加算手段を参照し前記各コードワードごとに当該コードワードについての前記歪みの累積値に応じて適応度を算出し前記適応度に応じて当該コードワードの複製を確率的に生成し、生成した前記各複製にそれぞれ微小なランダムベクトルを加算して新たなコードワードとし、前記各新たなコードワードをもって新たなコードブックとして前記コードブック記憶手段に保持されたコードブックを更新するものであることを特徴とするベクトル量子化器。

【請求項7】 有限個のコードワードからなるコードブックを保持するコードブック記憶手段と、逐次的に入力ベクトルが入力し前記入力ベクトルとの歪みが最小になるコードワードを前記コードブックの中から選択して最近傍ベクトルとする選択手段と、前記最近傍ベクトルに基づいて量子化結果を出力する量子化手段と、前記最近傍ベクトルと前記歪みに基づいて前記コードブック内のコードワードを逐次修正する修正手段とを有するベクトル量子化器において、

前記修正手段が、前記入力ベクトルの提示累積回数が所定数に達したときには、前記各コードワードごとに当該コードワードについての部分歪みに応じて適応度を算出し前記適応度に応じて確定的に当該コードワードを複製しあるいは消滅させ、生成した各複製にそれぞれ微小なランダムベクトルを加算して新たなコードワードとし、前記各新たなコードワードをもって新たなコードブックとして前記コードブック記憶手段に保持されたコードブックを更新するものであることを特徴とするベクトル量子化器。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、画像・音声データなどを圧縮して符号化する際に用いられるベクトル量子化器の設計方法と、この設計方法を応用したベクトル量子化器に関する。

【0002】

【従来の技術】画像データや音声データなどを圧縮して

$$Q: R^k \rightarrow Y$$

である。代表ベクトルを一般的に y_i で表わすことにすると、部分集合 Y は、 N 個の代表ベクトルからなる集合 $Y = \{y_i | i = 1, 2, \dots, N\}$ である。 y_i と Y は、それぞれコードワード、コードブックと呼ばれる。したがってこのようなベクトル量子化を行なうベクトル量子化器は、上述の k 次元ユークリッド空間 R^k を漏れなくかつ重複のない N 個の部分領域（部分空間） S_1, S_2, \dots, S_N に分割する。ここで、 $S_i = Q^{-1}(y_i)$ である。そして、入力ベクトルがどの部分領域に属するかによって、その属する部分領域に対応するコードワードにより入力ベクトルを近似し符号化する。

【0005】ベクトル量子化の性能は、入力ベクトルを

$$D = \sum_{i=1}^N D_i = \sum_{i=1}^N \int_{S_i} d(x, y_i) p(x) dx \quad (2)$$

ここで D_i は、コードワード y_i の支配領域 S_i (k 次元ユークリッド空間 R^k の部分空間のうちコードワード y_i に対応する部分空間) 内から生じる歪みを意味する。 $p(x)$ は、入力ベクトル x の確率分布である。歪み $d(x,$

$$d(x, y) = \|x - y\|^r$$

$$= \left(\sum_{j=1}^k |x_j - y_j|^2 \right)^{\frac{r}{2}} \quad (3)$$

が用いられる。

【0009】ところで、コードブックの学習手法として、従来から多くの手法が提案されている。前述したように、量子化器は部分領域の分割方法と各分割領域を代表する代表ベクトル値により特定される。最適コードブックの公知の学習指針として、(1) ボロノイ分割、(2) 重

符号化する方法の一つとして、ベクトル量子化 (vector quantization; VQ) がある。このベクトル量子化は、入力信号空間を離散的な信号で信号で近似する処理である。例えば音声データの場合、音声データを適宜の間隔でサンプリングして P 個のサンプルを得て、この P 個サンプルをまとめて P 次元の 1 個の入力ベクトルとする。予め適当な数だけ用意されている代表ベクトルの中から最もこの入力ベクトルに近いものを選択し、選択された代表ベクトルによって前記入力ベクトルを近似する。そして、選択された代表ベクトルを表わす番号をもって、入力ベクトルすなわち入力信号を符号化する方法である。ベクトル量子化は、データ圧縮や符号化の分野において、極めて重要な技術である。

【0003】ベクトル量子化は、形式的には、 k 次元ユークリッド空間 R^k から R^k の部分集合 Y への写像 Q として、定義される。すなわち、

【0004】

【数1】

(1)

それに対応するコードワードで近似する際の歪みで評価される。したがって最適なベクトル量子化を実現するためには、ベクトル量子化に先立って有限個の訓練ベクトルデータを用い、全てのコードワードにわたる平均歪み（歪みの統計的期待値）を最小化するように量子化器を設計しておかなければならない。この設計過程は、通常、コードブックの学習過程とも呼ばれる。

【0006】訓練ベクトルデータ x をコードワード y_i で近似したときの歪みを $d(x, y_i)$ とすると、平均歪み D は次式で定義される。

【0007】

【数2】

$y)$ としては、通常、ユークリッドノルムの r 乗：

【0008】

【数3】

心分割がある (Linde Y., Buzo A., and Gray R. M., "An algorithm Vector quantizer design", IEEE Trans. Commu., vol. 28, no. 1, 1980)。ところが、これらの条件は最適性のための弱い必要条件であるため、多くの局所最適値を有する。それゆえ、得られるコードブックの性能は代表ベクトルの初期値に大きく依存する。最

近、この問題を解決する方法として、“等確率原理”に基づく学習アルゴリズムが提案されている（例えば、Ahl t S, C., Krishnamurthy A. K., Chen P., and Melton D. E., “Competitive learning algorithm for vector quantization”, Neural Networks, 3, pp. 277-290, 1990), ここで等確率原理とは、各コードワードが等しい確率で最近傍ベクトルとなるようにするためのコードブックの設計原理である。しかし、等確率原理によった場合、上述した平均歪みDが最小化するという理論的保証は得られていない。

【0010】また本発明者らは、競合学習（Competitive Learning: CL）アルゴリズムに“増殖・消滅”機構を新たに付加したコードブック設計法を提案し（特願平4-294608）、競合学習に基づく従来の手法に対する優位性を示した。この特願平4-294608に示される方法では、学習過程において、各コードワードごとに、そのコードワードが最近傍ベクトルとなった頻度が記憶される。そして、予め定められた時間間隔で、記憶された頻度を基にしてコードワードの増殖・消滅が逐次実行される。この方法は、従来の競合学習アルゴリズムにおける“等確率原理”をより忠実に実現する方法である。

【0011】

【発明が解決しようとする課題】上述したように等確率原理による場合でも平均歪みDが最小値に到達するという保証はなく、平均歪みの大局的最小値を保証するコードブックの設計方法は、これまで知られていなかった。

【0012】本発明の目的は、ベクトル量子化における平均歪みを最小化するベクトル量子化器の設計方法と、この設計方法を応用したベクトル量子化器を提供することにある。

【0013】

【課題を解決するための手段】本発明の第1のベクトル量子化器の設計方法（請求項1, 2）は、予め有限個のコードワードからなるコードブックが用意され、ある確率分布にしたがう有限個の訓練ベクトルデータの各々に対し前記訓練ベクトルデータとの歪みが最小になるコードワードを前記コードブックの中から選択して最近傍ベクトルとする選択工程と、前記最近傍ベクトルと前記歪みに基づいて前記コードブック内のコードワードを逐次修正する修正工程とを有し、前記確率分布に対応する前記コードブックの設計を行なうベクトル量子化器の設計方法において、前記選択工程では、前記各コードワードの各々に対して、前記最近傍ベクトルとして選択された際の前記歪みの累積値を測定し、前記修正工程に、前記各コードワードごとに当該コードワードについての前記歪みの累積値に応じて適応度を算出し前記適応度に応じて当該コードワードの複製を生成する複製生成工程と、前記複製生成工程で生成した前記各複製にそれぞれ微小なランダムベクトルを加算して新たなコードワード

とし、前記各コードワードをもって新たな前記コードブックとする更新工程とが付加され、前記複製生成工程および前記更新工程とが、前記訓練ベクトルデータの提示累積個数が所定数に達したときに実行される。

【0014】本発明の第2のベクトル量子化器の設計方法（請求項3～5）は、予め有限個のコードワードからなるコードブックが用意され、ある確率分布にしたがう有限個の訓練ベクトルデータの各々に対し前記訓練ベクトルデータとの歪みが最小になるコードワードを前記コードブックの中から選択して最近傍ベクトルとする選択工程と、前記最近傍ベクトルと前記歪みに基づいて前記コードブック内のコードワードを逐次修正する修正工程とを有し、前記確率分布に対応する前記コードブックの設計を行なうベクトル量子化器の設計方法において、前記修正工程に、前記各コードワードごとに当該コードワードについての部分歪みに応じて適応度を算出し前記適応度に応じて当該コードワードを複製しあるいは消滅させる複製・消滅工程と、前記複製・消滅工程で生成した各複製にそれぞれ微小なランダムベクトルを加算して新たなコードワードとし、前記各コードワードをもって新たな前記コードブックとする更新工程とが付加され、前記複製・消滅工程および前記更新工程とが、前記訓練ベクトルデータの提示累積個数が所定数に達したときに実行される。

【0015】本発明の第1のベクトル量子化器（請求項6）は、有限個のコードワードからなるコードブックを保持するコードブック記憶手段と、逐次的に入力ベクトルが入力し前記入力ベクトルとの歪みが最小になるコードワードを前記コードブックの中から選択して最近傍ベクトルとする選択手段と、前記最近傍ベクトルに基づいて量子化結果を出力する量子化手段と、前記最近傍ベクトルと前記歪みに基づいて前記コードブック内のコードワードを逐次修正する修正手段とを有するベクトル量子化器において、前記各コードワードの各々について当該コードワードが前記最近傍ベクトルとして選択された際の前記歪みをそれぞれ積算して記憶する加算手段を有し、前記修正手段が、前記入力ベクトルの提示累積回数が所定数に達したときには、前記加算手段を参照し前記各コードワードごとに当該コードワードについての前記歪みの累積値に応じて適応度を算出し前記適応度に応じて当該コードワードの複製を確率的に生成し、生成した前記各複製にそれぞれ微小なランダムベクトルを加算して新たなコードワードとし、前記各新たなコードワードをもって新たなコードブックとして前記コードブック記憶手段に保持されたコードブックを更新するものである。

【0016】本発明の第2のベクトル量子化器（請求項7）は、有限個のコードワードからなるコードブックを保持するコードブック記憶手段と、逐次的に入力ベクトルが入力し前記入力ベクトルとの歪みが最小になるコー

ドワードを前記コードブックの中から選択して最近傍ベクトルとする選択手段と、前記最近傍ベクトルに基づいて量子化結果を出力する量子化手段と、前記最近傍ベクトルと前記歪みに基づいて前記コードブック内のコードワードを逐次修正する修正手段とを有するベクトル量子化器において、前記修正手段が、前記入力ベクトルの提示累積回数が所定数に達したときには、前記各コードワードごとに当該コードワードについての部分歪みに応じて適応度を算出し前記適応度に応じて確定的に当該コードワードを複製しあるいは消滅させ、生成した各複製にそれぞれ微小なランダムベクトルを加算して新たなコードワードとし、前記各新たなコードワードをもって新たなコードブックとして前記コードブック記憶手段に保持されたコードブックを更新するものであることを特徴とするベクトル量子化器。

【0017】

【作用】本発明によるコードブックの学習過程は、上述の特願平4-294608に示されるものと同様に“増殖・消滅”過程によるものであるが、増殖と消滅のための基準となる原理が根本的に異なっている。すなわち本発明では、上述の“等確率原理”ではなく、“等歪み原理”に基づいて、コードワードの増殖と消滅が行なわれる。等歪み原理とは、コードワード y_i から生ずる歪み D_i が全てのコードワードにわたってことごとく等しく($D_i = D/N$, $i = 1, 2, \dots, N$)となるようなコードブック設計原理である。

【0018】すなわち本発明の第1のベクトル量子化器の設計方法およびベクトル量子化器では、各コードワードごとに、そのコードワードが最近傍ベクトルとして選択されたときの歪みを積算しておく。そして、訓練ベクトルデータの累積提示回数が所定の値になったときに、歪みの累積値(累積歪量)の相対的な大小関係を基に“増殖”と“消滅”を行なって各コードワードの複製を作成し、得られた複製コードワードにノルムの小さなランダムノイズを加算する。累積歪量が大きかったコードワードに対してはその累積歪量に応じた数の複製が作成され、累積歪量の小さかったコードワードに対しては複製

が生成しないようにする。このように“増殖”と“消滅”を行なうことにより、コードワード y_i から生ずる歪み D_i が全てのコードワードにわたってことごとく等しくなる。この場合、コードワードの複製の作成は、確率的に行なわれる。通常の場合、コードワードの総数は複製の前後で変化しないようにする。また、訓練ベクトルデータの確率分布が、設計対象とするベクトル量子化器の入力対象データの確率分布と同様のものであることは、言うまでもない。

【0019】本発明の第2のベクトル量子化器の設計方法およびベクトル量子化器では、訓練ベクトルデータが全て提示されたときに、そのときの各コードワードごとの部分歪みの相対的な大小関係を基に“複製”と“消滅”を行なって各コードワードの複製を作成し、得られた複製コードワードにノルムの小さなランダムベクトルを加算する。部分歪みが相対的に大きかったコードワードに対しては理想的に多数の複製を作成し、部分歪みが小さかったコードワードに対してはそのコードワードを消滅させ、全体としてのコードワードの総数を保つようにする。以上の工程をある収束条件が満たされるまで繰り返す。したがって、有限個の訓練ベクトルデータが繰り返し用いられることになる。このように、最適値ベクトルの修正工程に、複製・消滅工程を付加することにより、コードワード y_i から生じる部分歪み D_i が全てのコードワードにわたって近似的に等しくなり、最適なコードワードが生成される。

【0020】ここで、この“等歪み原理”に基づくコードブック設計方法が最適であることを説明する。

【0021】Gershonは、確率分布 $p(x)$ が十分滑らか(連続)でありかつ十分大きな N (N はコードベクトルの数)に対して、平均歪み D^* の下限値が次の式を満たすことを理論的に示した(“Asymptotically optimal block quantization”, IEEE Trans. Inform. Theory, IT-25, 4, pp. 373-380, 1979)。

【0022】

【数4】

$$D^* \geq C(k, r) N^{-\beta} \|p(x)\|_{\frac{1}{1+\beta}} \quad (4)$$

ここで $\beta = r/k$ であり、 r はノルムの次数である。 $C(k, r)$ は量子化係数と呼ばれ、 k, r で決まる定数である。ただし、一般の k, r に対しては、 $C(k, r)$ の上限値のみが分かっているだけで、実際の値は未知である。 $C(k, r)$ は、以下の議論において重要ではないので、詳細は省略する。

【0023】

【外1】

ノルムと呼ばれ、次式で定義される。

【0024】

【数5】

$$\|p(x)\|_{\alpha} = \left(\int p(x)^{\alpha} dx \right)^{1/\alpha} \quad (5)$$

上記の下限値 D^* は、コードワード(代表ベクトル)の分布が

【外2】

$$\|p(x)\|_{\frac{1}{1+\beta}}$$

【0025】

に比例するときに得られ、これは若干の考察により、 $D_i = D^* / N$ が成立するときであることが分かる。

【0026】以上の議論は分布 $p(x)$ が連続な場合についてのものであったが、本発明では、一般の場合、すなわち分布が不連続な M 個のクラスからなる場合に、この議論を一般化する。各クラス内での信号分布を $p_j(x)$, $j = 1, \dots, M$ とし、 j 番目のクラス内には n_j の比率でコードワードが存在する場合を考える。式(3)に注意すると、全平均歪み D を最小化するための最適コ

ードワードの個数比率 n_j^* は、制約条件

【0027】

【数6】

$$\sum_{j=1}^M n_j = 1$$

のもとで、次の最小化問題を解くことにより得られる。

【0028】

【数7】

$$f(n_1, \dots, n_j) = \sum_{j=1}^M n_j^{-\beta} \|p_j(x)\|_{\frac{1}{1+\beta}} \rightarrow \min_{\{n_1, \dots, n_j\}} \quad (6)$$

この最小化問題は変分法により容易に解かれ、その解は以下ようになる。

【0029】

【数8】

$$n_j^* = \left(\|p_j(x)\|_{\frac{1}{1+\beta}} \right)^{\frac{1}{1+\beta}} / \sum_{j=1}^M \left(\|p_j(x)\|_{\frac{1}{1+\beta}} \right)^{\frac{1}{1+\beta}} \quad (7)$$

ゆえに、 j 番目のクラスに $(n_j^* N)$ 個のコードワードを配置し、かつ j 番目のクラス内の領域 $S_i^{(j)}$ での歪み $D_i^{(j)}$ がことごとく等しいときに、全てのクラスにわたる最小歪みが得られる。領域 $S_i^{(j)}$ での歪み $D_i^{(j)}$ がことごとく等しいとは、具体的には以下の場合である。

【0030】

$$\frac{D_i^{(j)}}{n_j^* N} = C(k, r) N^{-(1+\beta)} \left(\sum_{j=1}^M \int p_j(x)^{\frac{1}{1+\beta}} dx \right)^{1+\beta} \quad (8)$$

に注意すると、 $D_i^{(j)} / (n_j^* N)$ は j によらない定数であることが分かる。このことは、あるクラス内の各領域内での歪みが当該クラス内で等しいだけでなく、クラス間相互間でも等しいことを意味する。

【0032】以上の議論から、等歪み原理がコードブックの最適設計となることが示された。また、上記の解析から、従来手法である確率原理は、クラス内で一様分布となっている場合（ただし、クラス数がコードワードの数より十分小さい場合）にのみ最適となることがわかる。

【0033】ところで、本発明の特徴である上述の“等歪み原理”は、通常のベクトル量子化器のコードブックの最適設計に利用できるほか、適応ベクトル量子化法にも応用できる。“等歪み原理”を用いて適応ベクトル量子化を行なう場合には、訓練ベクトルデータのかわりに入力ベクトル（入力信号）そのものを対象として、上述の各工程を実施するようにすればよい。

【0034】また、本発明における“等歪み原理”は、通常のベクトル量子化器の最適設計にも利用できるほか、クラス分類型ベクトル量子化器の最適設計にも利用できる。ここでクラス分類型ベクトル量子化器とは、画像などのパタン認識において、入力画像の特徴を抽出し、抽出された特徴に基づいてクラス分けを行ない、クラスご

とに量子化を行なうものであって、予め定められている複数のクラスごとにコードブックが用意されている。本発明におけるクラス分類型ベクトル量子化器の設計方法は、上述した設計方法に対し、画像を同じサイズの矩形画像に分割してこの矩形画像を予め定められた複数の画像クラスに分類する分類工程と、分類された矩形画像に含まれる各画素値を列挙して得られるベクトルを訓練ベクトルデータとする訓練データ作成工程とが付加された構成となっている。この場合、部分領域の部分歪みを等しくするかわりに、分類工程により分類された第 j 番目のクラスの訓練データの個数比 P_j によって重み付けされた重み付き部分歪みを全てのコードワードにわたって等しくするように、複製・消滅工程を実施する。

【0035】

【実施例】

《第1の実施例》次に、本発明の実施例について、図面を参照して説明する。図1は、本発明の第1の実施例におけるベクトル量子化器の設計方法の実施に使用されるベクトル量子化器の構成を示すブロック図である。

【0036】このベクトル量子化器10は、ベクトルデータが入力する入力端子11、量子化結果を外部に出力するための出力端子12、入力端子11に接続され入力したベクトルデータを量子化して結果を出力端子12に

出力する量子化部 1、コードブック 3 を有するコードワード保持部 2、入力したベクトルデータを一定周期でサンプリングして訓練データとして出力する訓練データ生成部 4、歪計算部 5、歪加算テーブル 7 を内蔵した修正部 6 によって構成されている。ここでコードブック 3 は、予め定められた数 N のコードワード（代表ベクトル）を保持するものであり、その構成は図 2 に示す通りのものである。すなわち、各コードワードごとにその ID（識別番号）とそのコードワードのベクトル値（ベクトルの座標値、例えば k 次元ベクトルであるコードワード y_i に対して $(y_{i1}, y_{i2}, \dots, y_{ik})$ ）とを格納している。また歪計算部 5 は、訓練データ生成部 4 からの訓練データとコードブック 3 に格納されたコードワードの各々との歪み（例えばユークリッドノルムの r 乗）を計算し、歪みが最も小さいコードワードを最近傍ベクトルとし、そのコードワードの ID を修正部 6 と量子化部 1 とに送出するよう構成されている。また、選択されたコードワードと訓練データとの歪みは、歪計算部 5 から修正部 6 に送られる。

【0037】歪加算テーブル 7 は、図 3 に示すように、各コードワードごとに、その ID と累積歪量とを格納する。ここで累積歪量とは、そのコードワードが最近傍ベクトルとして選ばれた際の訓練データとそのコードワードとの歪みを積算したものである。修正部 6 は、歪計算部 5 からコードワードの ID と歪みの値とが送られてきた場合に、訓練データの入力数（コードワードの ID の 1 回の送出は、1 個の訓練データの入力に対応する）をカウント（累積）するとともに、歪加算テーブル 7 中のその ID に対応する累積歪量に今送られてきた歪みの値を加算し、従来の手法にしたがいコードワード保持部 2 にアクセスして最近傍ベクトルとなったコードワードを更新する。そして修正部 6 は、入力した訓練データの数がある値になったとき、①歪加算テーブル 7 に記憶された各コードワードごとの累積歪量にしたがって、各コードワードごとに適応度を算出し、②適応度にしたがって各コードワードの複製を生成し、複製されたコードワードにノルムが微小なランダムベクトルを加算し、得られたコードワード群を持ってコードブック 3 を書き換え、③

歪加算テーブル 7 を初期化（各累積歪量を 0 にする）し、次に複製生成を行なうこととなる入力ベクトルの累積数に、前記ある値を書き換えるように構成されている。適応度とは、累積歪量の相対的な大小関係のことを示している。

【0038】次に、本実施例の動作を図 4 のフローチャートを用いて説明する。

【0039】コードブック 3 には、予め定められた数（例えば N 個）のコードワードが、乱数で初期化されて記憶されている。

【0040】ベクトルデータが入力すると、そのベクトルデータは、量子化部 1 と訓練データ生成部 4 とに送られ、訓練データ生成部 4 で一定周期ごとにサンプリングされ、訓練データ $x(t)$ として歪計算部 5 に送られる。歪計算部 5 は、コードブック 3 内のコードワード y_i ($1 \leq i \leq N$) のそれぞれと入力した訓練データ $x(t)$ との歪みを計算し、この歪みが一番小さいコードワードを最近傍ベクトル y_c として決定する（ステップ 101）。すなわち、

【0041】

【数 11】

を満たすものが最近傍ベクトル y_c として選び出される。そして選び出されたコードワードの ID が量子化部 1 と修正部 6 に送られ、歪み $d(x, y_c)$ の値が修正部 6 に送られる。量子化部 1 は、送られてきた ID に基づいて、量子化データを外部に出力する。

【0042】一方、修正部 6 は、訓練データの累積入力数 t をカウントしており、 t がある値 T と一致しているかどうかをまず調べる（ステップ 102）。

【0043】 $t \neq T$ である場合には、歪加算テーブル 7 において入力した ID に対応する累積歪量 g_c に歪み $d(x, y_c)$ を加算する（ステップ 103）。そして、次式にしたがって、最近傍ベクトル y_c に対応するコードワードを修正・更新してコードブック 3 に格納する（ステップ 104）。

【0044】

【数 12】

ここで $\varepsilon(t)$ は、訓練データの累積入力数 t に対する時間に関する減少関数である。

【0045】一方、ステップ 102 で $t = T$ であった場合、修正部 6 は歪加算テーブル 7 を参照して、各コードワードごとの累積歪量 g_i ($1 \leq i \leq N$) を正規化する

$$\hat{g}_i = \frac{g_i}{\sum_{i=1}^N g_i}, \quad \text{for } i=1, 2, \dots, N \quad (10)$$

続いて修正部 6 は、実数からなる区間 $[0, 1]$ をステップ 105 で得られた正規化された累積歪量

の正規化は、次式にしたがって行なわれる。

【0046】

【数 13】

【0047】

【外 3】

$$\hat{g}_i, \quad i=1,2,\dots,N$$

によって、分割する。これにより区間 $[0, 1]$ は、長さ累積歪量 g_i に比例した N 個の部分区間 (I_1, I_2, \dots, I_N) に分割される。続いて各コードワードに対する複製個数 (m_i とする) を初期化する ($m_i=0, i=1, \dots, L$)。そして区間 $[0, 1]$ に対する一様乱数を N 個発生させ、その各々の乱数 ($\in [0, 1]$) に対し、上述の部分区間 ($I_i=0, i=1, \dots, N$) のどの区間に属するかを判定する。例えば I_j に属する場合、その区間に対応する複製個数に 1 を加算する ($m_i \leftarrow m_i + 1$)。これにより、コードワード y_i の複製後の個数 m_i が前記累積歪量 g_i に確率的に比例して得られることになる。明らかに

【0048】

【数14】

$$y_i^*(t) = y_i(t) + \delta_i(t)$$

ここで $\delta_i(t) \in R^k$ は、 $\|\delta_i(t)\| \ll \|y_i(t)\|$ を満たすノルムの微小なランダムベクトルである。以上のステップ105～ステップ107によって、各コードワード近傍に、前記累積歪量に比例した複製が確率的に配置されるから、この新たに生成した複製でもって、コードブック3全体を更新する(ステップ108)。

【0051】そして、修正部6は、全コードワードに対して歪加算テーブル7を初期化し(ステップ109)、上述の値 T に $2T$ を代入して(ステップ110)、処理を終了する。ここで T に $2T$ を代入したことにより、入力する訓練データの累積個数が $2T$ に達した時点で再び上記のステップ105～ステップ110が行なわれることになる。

【0052】ここで、複製過程について、実例を挙げて説明する。ここでは、コードワードの数が7個であって、訓練データの入力数が所定の数に達したときに、累積頻度から適応度を求めて複製を作る場合の処理を取り上げる。例えば、この時点で7個のコードワード y_i ($1 \leq i \leq 7$) のそれぞれの累積歪量 g_i が図5の(A)に示すものであったとする。これに対する新たなコードワード y_i^* が図5の(B)に示されている。累積歪量が0.0であったコードワード m_3, m_4, m_7 に対しては複製が作成されず、累積歪量が40.2であって一番大きかったコードワード m_1 に対しては3個の複製が作成されていることがわかる。

【0053】次に、同一の訓練データに対するコードブックの学習過程について、本実施例の方法に基づくものと、等確率原理に基づく従来の方法に基づくものとを比較した結果を説明する。実験で使用した訓練データとしては、図6(a)に示す不連続な1次元分布を使用した。この分布は、区間 $[0.0, 1.0]$ の間の任意の値をとる2つの一様乱数の積を計算し、計算で得た分布から区

$$\sum_i m_i = N$$

が成立するから、複製後のコードワードの総数は複製前と変わらない。そして、各コードワード y_i ごとに、そのコードワード y_i に対する複製個数 m_i だけの複製の代表ベクトルを作成する(ステップ106)。この時点では、複製されたコードワードは、対応するもとのコードワードと同じである。

【0049】次に、複製された各コードワードに対し、以下の式によって微小なノルムのランダムベクトルをそれぞれ加算して、新たなコードワードを設定する(ステップ107)。ここで y_i^* (ただし $i=1, \dots, N$) は、新たなコードワードの座標値である。

【0050】

【数15】

$$(11)$$

間 $[0.2, 0.4]$ に属するものを取り除くことによって得た。訓練データ数は7000とした。また、コードワードの総数 N は15とし、各コードワードの初期値を一律に0.35とした。図6(b)に示したものは本実施例による学習過程を示し、図6(c)は従来の方法による学習過程を示している。これらの図で、縦軸は時間の推移すなわち訓練データの累積入力数に対応し、横軸は、各コードワードの値に対応する。

【0054】ところで、1次元の場合には、式(2)の平均歪み D の最小値とそれに対応するコードワード y_i の最適値を数値的探索法により求めることができ (Max J., "Quantizing for minimum distortion", IRE Trans. Inform. Theory, IT-6, pp. 7-12, 1960)、理論値との比較が可能となる。図6(a), (b)において、図中の点線は数値的に求めた理論値を示す。

【0055】本実施例による結果を示す図6(b)では、コードワードの値が理論値に漸近していく様子を確認できる。また、学習終了後、同じデータに対する歪み d (歪みとしては自乗誤差を採用) を算出したところ、理論値の2%以下の誤差という極めて良好な結果が得られた。一方、従来の方法による結果を示す図6(c)では、コードワードの収束値と理論値との差がかなりあり、等確率原理が必ずしも最適ではないことを示している。なお、本実施例に基づく図6(b)において、分岐や消滅が見られるが、これは上述した増殖・消滅によるものである。

【0056】《第2の実施例》次に、本発明の第2の実施例について説明する。図7は本発明の第2の実施例におけるベクトル量子化器の設計方法の実施に使用されるベクトル量子化器の構成を示すブロック図である。

【0057】このベクトル量子化器20は、図1に示した第1の実施例でのベクトル量子化器10と比べ、歪計

算部25と修正部26の構成が異なっている。図7において、図1に示したものと同一ブロックには、同一の参照符号が付されている。以下では、第1の実施例との相違点を中心に説明を行なう。

【0058】歪計算部25は、訓練データ生成部4からの訓練データとコードブック3に格納されたコードワードの各々との歪み（例えばユークリッドノルムの r 乗）を計算し、歪みが最も小さいコードワードを最近傍ベクトルとし、そのコードワードのIDを修正部26と量子化部1とに送出し、選択されたコードワードと訓練データとの歪みを修正部26に送るように構成されている。さらに、歪計算部25は、全ての訓練データを用いてコードワードが更新されたとき、再度、全ての訓練データを用いて現在のコードワードの値を用いて歪み（部分歪み）を算出し、各コードワードごとの部分歪みを修正部26に送出するようになっている。

【0059】修正部26には、部分歪みテーブル27が設けられている。部分歪みテーブル27は、図8に示すように、各コードワードごとに、そのIDと歪計算部25から送られてきた部分歪みとを格納する。修正部26は、歪み計算部25から最近傍ベクトルに対応するコードワードのIDと歪みの値が送られてきた場合に、従来の手法にしたがいコードワード保持部2にアクセスして最近傍ベクトルとなったコードワードを更新する（例えば上述の式(9)）。そして修正部26は、全ての訓練データを用いてコードワードが更新されたときに歪計算部25から各コードワードごとの部分歪みが送られてくると、①各コードワードごとの部分歪みを部分歪みテーブル27に格納し、②部分歪みテーブル27内に格納された各コードワードごとの部分歪みに応じて各コードワードごとに適応度を算出し、③適応度にしたがって後述するように各コードワードを複製しあるいは消滅させ、

複製されたコードワードにノルムが微小なランダムベクトルを加算し、④得られたコードワード群をもってコードブック3を書き換えるように構成されている。

【0060】次に、本実施例の動作について図9のフローチャートを用いて説明する。

【0061】コードブック3には予め定められた数のコードワード（代表ベクトル）が記憶されているとする。そしてベクトルデータが入力すると、第1の実施例と同様に訓練データ $x(t)$ として歪計算部25に送られ、最近傍ベクトル y_e が決定される（ステップ201）。修正部26は、上記の式(9)にしたがって最近傍ベクトル y_e に対応するコードワードを修正・更新し、コードブック3に格納する（ステップ202）。そして全ての訓練データに基づいてコードワードが更新されたかどうかを判定される（ステップ203）。全ての訓練データに基づいてコードワードが更新された場合には、ステップ204に移行する。

【0062】ステップ204では、歪計算部25によって、全ての訓練データを用いて現在のコードワード値に対する部分歪みが算出され、この部分歪みはコードワードごとに部分歪みテーブル27に格納される。続いて、各コードワードごとに、当該コードワードの部分歪み D_j を基に、次式によって正規化適応度 h_j が算出される（ステップ205）。ここで j は j 番目のコードワードであることを示している。 γ (< 1.0) は非負の定数であって、これにより極端な淘汰が抑制される。また、 $s(m)$ は、 m 回目の繰り返しにおける複製・消滅対象コードワード数を表わすものであって、 m とともに0にまで減少する関数とする。これにより、複製・消滅処理は有限回で必ず終了する。

【0063】

【数16】

$$h_j = D_j^\gamma / \sum_{j=1}^{s(m)} D_j^\gamma \quad (12)$$

続いて、以下の手順(I), (II)による確定的な淘汰を未処理のコードワードに対する複製個数 (u_j ; $j = 1, 2, \dots, N$) を決定する（ステップ206）。

【0064】手順(I) : j 番目のコードワード y_j に対し、 $[h_j \cdot s(m)]$ ($= u_j$) を計算する。ただし、 $[a]$ は a を越えない最大の整数とする。

【0065】手順(II) : 全てのコードワード y_j ($j = 1, 2, \dots, N$) に対し、 $h_j \cdot s(m)$ の値の大きなものから順に、

【0066】

【外4】

$$s(m) = \sum_{j=1}^{s(m)}$$

個のコードワードを選び、選択されたコードワードに対

応する複製個数 (u_j) に1を加算する。

【0067】以上のようにコードワードごとの複製個数が求められたら、複製個数が0すなわち $u_j = 0$ であるコードワードを消滅させて、以下の処理の処理対象から除外する（ステップ207）。そして、各コードワード y_j の複製個数 u_j にしたがい、各コードワードに対し、 $\|\delta_{j,1}\| \ll \|y_j\|$ なるランダムな摂動ベクトル $\delta_{j,1}$ （ただし $l = 1, 2, \dots, u_j - 1$ ）を加算する（ステップ208）。これにより、コードワード y_j の近傍に計 $u_j - 1$ 個の重みベクトル（複製）が生成されたことになり、各コードワードの近傍にその部分歪みの相対値に比例した複製が配置されたことになる。このようにして得られた複製を新たなコードワードとし、複製の親となったコードワードと新たに生成されたコードワードとをもって、コードブック3全体を更新する（ステップ20

9)。重みベクトルの個数が各コードワード y_j に対して u_j-1 個であることにより、明らかに、複製後のコードワード全体の個数は複製前と変わらない。以上のようにして、本実施例における複製・消滅過程が終了したことになる。

【0068】ここで、複製・消滅過程について実例を挙げて説明する。ここでは7個のコードワードについて、それぞれの部分歪みが図10(A)に示しようなものであったとし、 $s(m)=7$ であるとする。これに対する新たなコードワードが図10(B)に示されており、部分歪みの相対値に比例した複製が生成していることが分かる。

【0069】次に、同一の訓練データに対するコードブックの学習過程について、本実施例の方法に基づくものと、等確率原理に基づく従来の方法に基づくものとを比較した結果を説明する。実験で利用した訓練データとしては、図11(a)に示す不連続な1次元分布を使用した。この分布は、区間 $[0.0, 1.0]$ の間の任意の値をとる2つの一様乱数の積を計算し、計算で得た分布から区間 $[0.2, 0.4]$ に属するものを取り除くことによって得た。訓練データ数は7000とした。また、コードワードの総数Nは15とし、各コードワードの初期値を一律に0.35とした。図11(b)は本実施例による学習過程を示している。図11(c)は従来の方法による学習過程を示しており、上述の第1の実施例で図6(c)に示したものと同一である。これらの図で、縦軸は時間の推移すなわち訓練データの累積入力数に対応し、横軸は、各コードワードの値に対応する。また、図11(a)、(b)において図中の点線は、図6(a)、(b)におけるの場合と同様に数値的に求めた理論値を示す。

【0070】本実施例による結果を示す図11(b)では、コードワードの値が理論値に漸近していく様子を確認できる。また、学習終了後、同じデータに対する歪み d (歪みとしては自乗誤差を採用)を算出したところ、理論値の2%以下の誤差という極めて良好な結果が得られた。一方、従来の方法による結果を示す図11(c)では、コードワードの収束値と理論値との差がかなりあり、等確率原理が必ずしも最適ではないことを示している。なお、本実施例に基づく図11(b)において、分岐や消滅が見られるが、これは上述した増殖・消滅によるものである。

【0071】

【発明の効果】以上説明したように本発明は、等歪み原理に基づいて、コードワードの増殖および消滅処理を行なうことにより、分布の如何を問わず、平均歪みを最小

かできる最適なコードブックを設計できるという効果がある。

【図面の簡単な説明】

【図1】本発明の第1の実施例におけるベクトル量子化器の構成を示すブロック図である。

【図2】コードブックの構成例を示す図である。

【図3】歪加算テーブルの構成例を示す図である。

【図4】図1に示したベクトル量子化器の動作を説明するフローチャートである。

【図5】第1の実施例におけるコードワードの複製過程を説明する図である。

【図6】(a)は第1の実施例における訓練データの確率密度分布図、(b)は(a)に示す確率密度分布を示す訓練データに対して本発明の方法によりコードワードの学習を行なった状況を示す図、(c)は(a)に示す確率密度分布を示す訓練データに対して従来の方法によりコードワードの学習を行なった状況を示す図である。

【図7】本発明の第2の実施例におけるベクトル量子化器の構成を示すブロック図である。

【図8】部分歪みテーブルの構成例を示す図である。

【図9】図7に示したベクトル量子化器の動作を説明するフローチャートである。

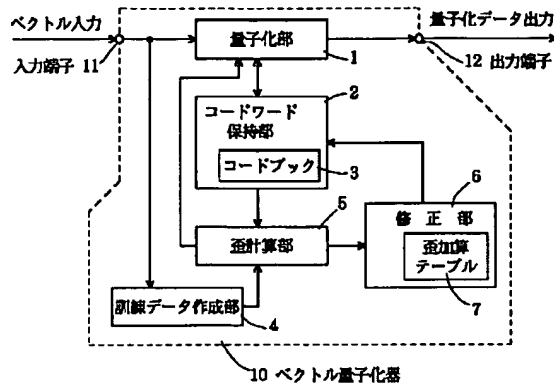
【図10】第2の実施例におけるコードワードの複製・消滅過程を説明する図である。

【図11】(a)は第2の実施例における訓練データの確率密度分布図、(b)は(a)に示す確率密度分布に対して本発明の方法によりコードワードの学習を行なった状況を示す図、(c)は(a)に示す確率密度分布を示す訓練データに対して従来の方法によりコードワードの学習を行なった状況を示す図である。

【符号の説明】

- 1 量子化部
- 2 コードワード保持部
- 3 コードブック
- 4 訓練データ作成部
- 5, 25 歪計算部
- 6, 26 修正部
- 7 歪加算テーブル
- 10, 20 ベクトル量子化器
- 11 入力端子
- 12 出力端子
- 27 部分歪みテーブル
- 101~110, 201~209 ステップ

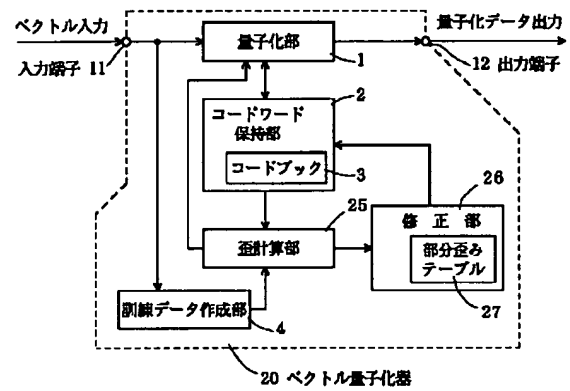
【図1】



【図2】

ID	ベクトル値
1	$(y_{11}, y_{12}, \dots, y_{1k})$
2	$(y_{21}, y_{22}, \dots, y_{2k})$
3	$(y_{31}, y_{32}, \dots, y_{3k})$
\vdots	\vdots
N	$(y_{N1}, y_{N2}, \dots, y_{Nk})$

【図7】



【図3】

ID	累積歪量
1	δ_1
2	δ_2
3	δ_3
\vdots	\vdots
N	δ_N

【図5】

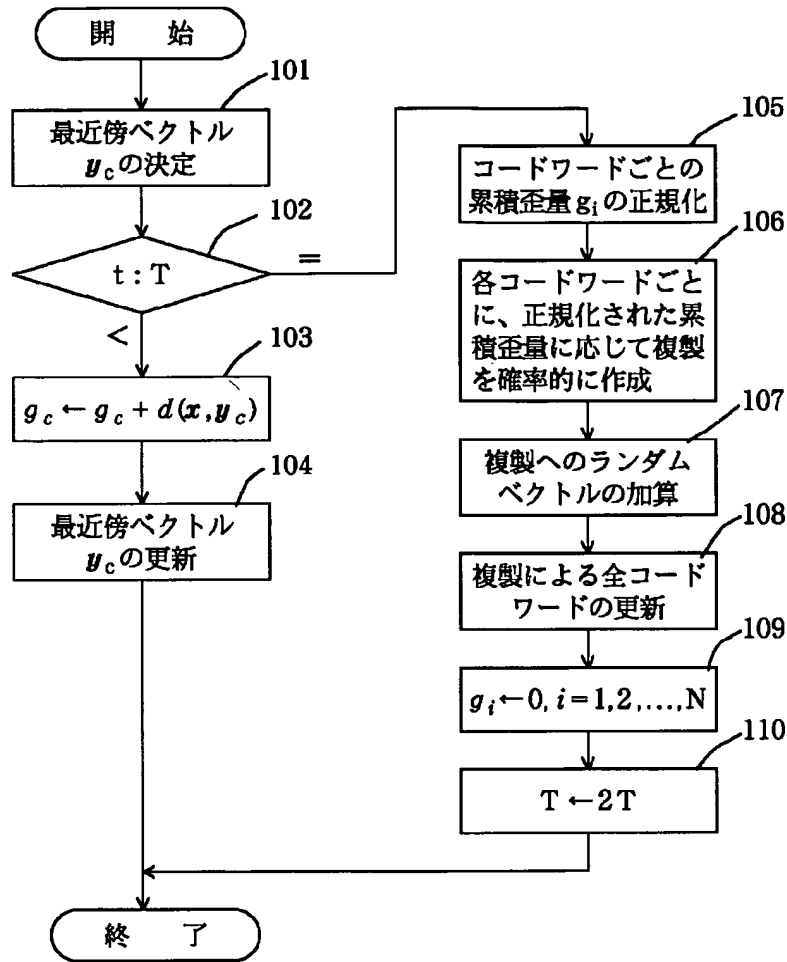
ID	コードワード	累積歪量
1	y_1	40.2
2	y_2	25.6
3	y_3	0.0
4	y_4	0.0
5	y_5	30.3
6	y_6	14.8
7	y_7	0.0

(A)

新しいID	新しいコードワードの内容
1	$y_1 + \delta_1$
2	$y_1 + \delta_2$
3	$y_1 + \delta_3$
4	$y_2 + \delta_4$
5	$y_5 + \delta_5$
6	$y_5 + \delta_6$
7	$y_7 + \delta_7$

(B)

【図4】

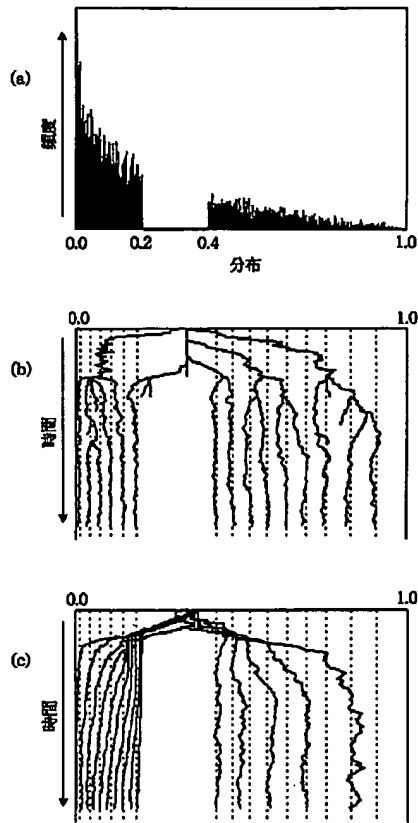


【図8】

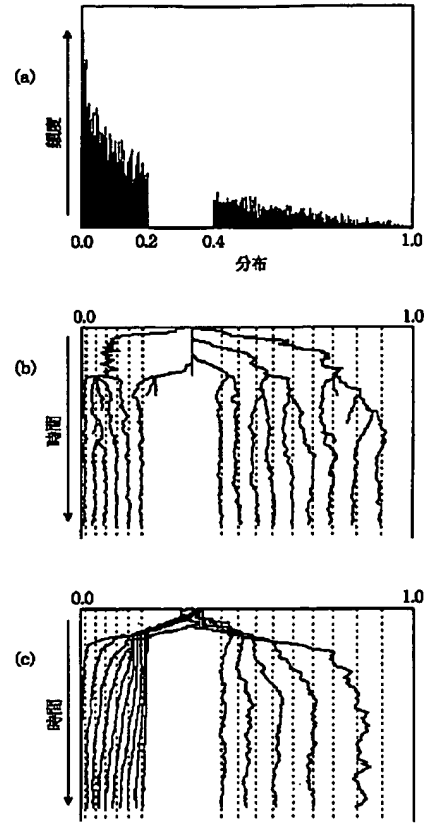
27 部分歪みテーブル

ID	部 分 歪 み
1	D ₁
2	D ₂
3	D ₃
⋮	⋮
N	D _N

【図 6】



【図 11】



【図 10】

ID	コード ワード	部分歪み
1	y_1	40.2
2	y_2	25.6
3	y_3	0.0
4	y_4	0.0
5	y_5	30.3
6	y_6	14.8
7	y_7	0.0

(A)



新たなID	新たなコード ワードの値
1	$y_1 + \delta_1$
2	$y_1 + \delta_2$
3	$y_1 + \delta_3$
4	$y_2 + \delta_4$
5	$y_5 + \delta_5$
6	$y_5 + \delta_6$
7	$y_7 + \delta_7$

(B)

【図9】

